Foundation of Cryptography, Lecture 10
Secure Computation, *Two* Parties[1]

Handout Mode

Iftach Haitner

Tel Aviv University.

December 31, 2025

[1] Prepared: 2025/12/29,20:02:34.

Section 1

The Model

Multiparty Computation

- Multiparty Computation – computing a functionality f
- Secure Multiparty Computation: compute f in a “secure manner"
  - Correctness
  - Privacy
  - Independence of inputs
  - Guaranteed output delivery
  - Fairness : corrupted parties should get their output iff the honest parties do
  - and ...
- Examples: coin-tossing, broadcast, electronic voting, electronic auctions
- How should we model it?
- Real Vs. Ideal paradigm

Real-model execution

For a protocol $\pi = (\mathsf{A}, \mathsf{B})$ and inputs $x_c, x_\mathsf{A}, x_\mathsf{B} \in \{0,1\}^*$, let $\mathsf{REAL}_\pi(x_c, x_\mathsf{A}, x_\mathsf{B})$ be the joint output of $(\mathsf{A}(x_\mathsf{A}), \mathsf{B}(x_\mathsf{B}))(x_c)$.

Given a two-party protocol π, an algorithm taking the role of one of the parties in π is:

- Honest. Acts exactly according to π.
- Semi-honest. Acts honestly but (1) Might prematurely abort (2) Outputs its *view*.
- Malicious. Acts arbitrarily.

π' is admissible with respect to π, if at least one party is honest.

Ideal model execution

For a pair of oracle-aided algorithms $\widehat{\pi} = (\widehat{\mathsf{A}}, \widehat{\mathsf{B}})$, inputs $x_c, x_{\widehat{\mathsf{A}}}, x_{\widehat{\mathsf{B}}} \in \{0,1\}^*$ and a function $f = (f_{\widehat{\mathsf{A}}}, f_{\widehat{\mathsf{B}}})$, let $\mathrm{IDEAL}^f_{\widehat{\pi}}(x_c, x_{\widehat{\mathsf{A}}}, x_{\widehat{\mathsf{B}}})$ be the joint output of the parties in the end of the following experiment:

1. The input of $\widehat{\mathsf{P}}$, $\widehat{\mathsf{P}} \in \{\widehat{\mathsf{A}}, \widehat{\mathsf{B}}\}$, is $(x_c, x_{\widehat{\mathsf{P}}})$.
2. $\widehat{\mathsf{P}}$ sends $y_{\widehat{\mathsf{P}}}$ to the trusted party .
3. Trusted party sends $z_{\widehat{\mathsf{P}}} = f_{\widehat{\mathsf{P}}}(y_{\widehat{\mathsf{A}}}, y_{\widehat{\mathsf{B}}})$ to $\widehat{\mathsf{P}}$ in an arbitrary order.
   - after receiving its output, a party can instruct trusted party to abort: not send the output to other party.
4. Each party outputs some value.

An oracle-aided algorithm is:

- Honest. Sends its private input to the trusted party (i.e., sets $y_{\widehat{\mathsf{P}}} = x_{\widehat{\mathsf{P}}}$), and its only output is the value it gets from the trusted party (i.e., $z_{\widehat{\mathsf{P}}}$).
- Semi-honest. Acts honestly but (1) Might ask the trusted party to abort (2) Outputs its view.
- Malicious. Acts arbitrarily.

$\widehat{\pi}$ is admissible, if at least one party is honest.

Secure computation

Definition 1 (Secure computation)

A PPT protocol $\pi = (\mathsf{A}, \mathsf{B})$ securely computes f, if $\forall$ admissible PPT protocol $\pi' = (\mathsf{A}', \mathsf{B}')$, exists admissible PPT pair $\widehat{\pi} = (\widehat{\mathsf{A}}, \widehat{\mathsf{B}})$, s.t.

$$\{\mathrm{REAL}_{\pi'}(x_c, x_{\mathsf{A}}, x_{\mathsf{B}})\}_{x_c, x_{\mathsf{A}}, x_{\mathsf{B}} \in \{0,1\}^*} \approx_c \{\mathrm{IDEAL}^f_{\widehat{\pi}}(x_c, x_{\mathsf{A}}, x_{\mathsf{B}})\}_{x_c, x_{\mathsf{A}}, x_{\mathsf{B}} \in \{0,1\}^*}$$

In case π' is honest, we require that $\widehat{\pi}$ is honest, and the ensembles to be identical.

- Recall that the enumeration index (i.e., $x_c, x_{\mathsf{A}}, x_{\mathsf{B}}$) is given to the distinguisher.
- π securely computes f implies that π computes f correctly.
- Security parameter
- Auxiliary inputs
- We start with presenting a semi-honest secure protocol for any two-party functionality, and then compile it into a maliciously secure protocol.

Section 2

Coin flipping

Coin flipping

An coin-flipping protocol securely computes the functionality $\mathsf{CF}(1^\ell) = (U_\ell, U_\ell)$ for $U_\ell \stackrel{R}{\leftarrow} \{0,1\}^\ell$.

- We will focus on single bit.

The protocol

- Let Com be a perfectly binding, NI, cmt.
- Let $(\mathsf{P}^1, \mathsf{V}^1)$ be a ZKP-POK for $\mathcal{L}^1 = \{((1^n, c), (b, r)) \colon c = \mathsf{Com}(1^n, b; r)\}$.
- Let $(\mathsf{P}^2, \mathsf{V}^2)$ be a ZKP-POK for $\mathcal{L}^2 = \{((1^n, c, b), r) \colon c = \mathsf{Com}(1^n, b; r)\}$.

Protocol 2 (Coin flipping (A, B))

Common input: 1^n.
In parallel (for both $\mathsf{C} \in \{\mathsf{A}, \mathsf{B}\}$):

1. C Sample $b_\mathsf{C} \stackrel{R}{\leftarrow} \{0,1\}$ and $r_\mathsf{C} \stackrel{R}{\leftarrow} \{0,1\}^n$, and send $c_\mathsf{C} \leftarrow \mathsf{Com}(b_\mathsf{C}; r_\mathsf{C})$.
2. The parties interact in $(\mathsf{P}^1(b_\mathsf{C}, r_\mathsf{C}), \mathsf{V}^1)(1^n, c_\mathsf{C})$.
3. C: Reveal b_C.
4. The parties interact in $(\mathsf{P}^2(r_\mathsf{C}), \mathsf{V}^2)(1^n, c_\mathsf{C}, b_\mathsf{C})$.
5. C: Output $b_\mathsf{A} \oplus b_\mathsf{B}$.

Claim 3

Protocol 2 securely computes CF (with abort).

Proving Claim 3

We need to prove that $\forall$ (admissible) PPT protocol $\pi' = (\mathsf{A}', \mathsf{B}')$ for (A, B), exists oracle-aided PPT pair $\widehat{\pi} = (\widehat{\mathsf{A}}, \widehat{\mathsf{B}})$ s.t.

$$\{\mathrm{REAL}_{\pi'}(1^n)\} \approx_c \{\mathrm{IDEAL}^{\mathsf{CF}}_{\widehat{\pi}}(1^n)\}, \tag{1}$$

where the enumeration is over $n \in \mathbb{N}$.

Malicious A'

Let S be the (BB) simulator for $(\cdot, \mathsf{V}_2)$. Define the malicious oracle-aided strategy $\widehat{\mathsf{A}}$ as follows:

Algorithm 4 ($\widehat{\mathsf{A}}$)

input: 1^n.

1. Call CF, let b be the output.
2. Emulate $(\mathsf{A}', \mathsf{B})(1^n)$ for the first two steps, and extract b_{A} from A'.
3. Continue the emulation till its end, while changing b_{B} to $b \oplus b_{\mathsf{A}}$, and simulating A''s view in the execution of $(\mathsf{P}^2, \mathsf{V}_2)$, where it acts as the verifier, using $\mathsf{S}_2^{\mathsf{A}'}(1^n, c_{\mathsf{B}}, b_{\mathsf{B}})$.
4. Abort CF if the executions aborts.

Let $\pi' = (\mathsf{A}', \mathsf{B})$ and $\widehat{\pi} = (\widehat{\mathsf{A}}, \widehat{\mathsf{B}})$, where $\widehat{\mathsf{B}}$ is honest.

Claim 5

$\{\mathrm{REAL}_{\pi'}(1^n)\} \approx_c \{\mathrm{IDEAL}^{\mathsf{CF}}_{\widehat{\pi}}(1^n)\}$.

Proof?

Section 3

Oblivious Transfer

Oblivious transfer

An (one-out-of-two) OT protocol securely computes the functionality $\mathsf{OT} = (\mathsf{OT}_\mathsf{S}, \mathsf{OT}_\mathsf{R}))$ over $(\{0,1\}^* \times \{0,1\}^*) \times \{0,1\}$, where $\mathsf{OT}_\mathsf{S}(\cdot) = \perp$ and $\mathsf{OT}_\mathsf{R}((\sigma_0, \sigma_1), i) = \sigma_i$.

- “Complete" for multiparty computation
- We show how to construct for bit inputs.

Oblivious transfer from trapdoor permutations

Let (G, f, Inv) be a TDP and let b be an hardcore predicate for f.

Protocol 6 ((S, R))

Common input: 1^n
S**'s input:** $\sigma_0, \sigma_1 \in \{0,1\}$.
R**'s input:** $i \in \{0,1\}$.

1. S chooses $(e,d) \leftarrow \mathsf{G}(1^n)$, and sends e to R.
2. R chooses $x_0, x_1 \leftarrow \{0,1\}^n$, sets $y_i = f_e(x_i)$ and $y_{1-i} = x_{1-i}$, and sends y_0, y_1 to S.
3. S sets $c_j = b(\mathrm{Inv}_d(y_j)) \oplus \sigma_j$, for $j \in \{0,1\}$, and sends (c_0, c_1) to R.
4. R outputs $c_i \oplus b(x_i)$.

Claim 7

Protocol 6 securely computes OT (in the semi-honest model).

Proving Claim 7

We need to prove that $\forall$ semi-honest (admissible) PPT protocol $\pi' = (\mathsf{S}', \mathsf{R}')$ for (S, R), exists oracle-aided PPT pair $\widehat{\pi} = (\widehat{\mathsf{S}}, \widehat{\mathsf{R}})$ s.t.

$$\{\mathsf{REAL}_{\pi'}(1^n, (\sigma_0, \sigma_1), i)\} \approx_c \{\mathsf{IDEAL}^{\mathsf{OT}}_{\widehat{\pi}}(1^n, (\sigma_0, \sigma_1), i)\}, \tag{2}$$

where the enumeration is over $n \in \mathbb{N}$ and $\sigma_0, \sigma_1, i \in \{0, 1\}$.

R's security

For a semi-honest S', define semi-honest oracle-aided strategy $\widehat{\mathsf{S}}$ as follows:

Algorithm 8 ($\widehat{\mathsf{S}}$)

input: $1^n, \sigma_0, \sigma_1$

1. Send (σ_0, σ_1) to the trusted party.
2. Emulate $(\mathsf{S}'(1^n, \sigma_0, \sigma_1), \mathsf{R}(1^n, 0))$.
3. Output the output that S' does.

Let $\pi' = (\mathsf{S}', \mathsf{R})$ and $\widehat{\pi} = (\widehat{\mathsf{S}}, \widehat{\mathsf{R}})$, where $\widehat{\mathsf{R}}$ is honest.

Claim 9

$\{\mathsf{REAL}_{\pi'}(1^n, (\sigma_0, \sigma_1), i)\} \equiv \{\mathsf{IDEAL}^{\mathsf{OT}}_{\widehat{\pi}}(1^n, (\sigma_0, \sigma_1), i)\}$.

Proof?

S's security

For a semi-honest implementation R' of R, define the oracle-aided semi-honest strategy $\widehat{\mathsf{R}}$ as follows.

Algorithm 10 ($\widehat{\mathsf{R}}$)

input: $1^n, i \in \{0,1\}$,

1. Send i to the trusted party, and let σ be its answer.
2. Emulate $(\mathsf{S}(1^n, \sigma_0, \sigma_1), \mathsf{R}'(1^n, i))$, for $\sigma_i = \sigma$ and $\sigma_{1-i} = 0$.
3. Output the output that R' does.

Let $\pi' = (\mathsf{S}, \mathsf{R}')$ and $\widehat{\pi} = (\widehat{\mathsf{S}}, \widehat{\mathsf{R}})$, where $\widehat{\mathsf{S}}$ is honest.

Claim 11

$\{\mathrm{REAL}_{\pi'}(1^n, (\sigma_0, \sigma_1), i)\} \approx_c \{\mathrm{IDEAL}^{\mathrm{OT}}_{\widehat{\pi}}(1^n, (\sigma_0, \sigma_1), i)\}$.

Proof?

Section 4

Yao Garbled Circuit

Before we start

- Fix a (multiple message) semantically-secure private-key encryption scheme $(\mathsf{G},\mathsf{E},\mathsf{D})$ with
  1. $\mathsf{G}(1^n) = U_n$.
  2. For any $m \in \{0,1\}^*$ $\Pr_{(d,d') \leftarrow (\{0,1\}^n)^2}[\mathsf{D}_d(\mathsf{E}_{d'}(m)) \neq \perp] = \mathsf{neg}(n)$.
- Can we construct such a scheme?

  Yes, append 0^n at the end of the message.

  Or use the MAC-based CCA2 scheme
- Boolean circuits: gates, wires, inputs, outputs, values, computation

The Garbled Circuit

Fix a Boolean circuit C and $n \in \mathbb{N}$.

- Let $\mathcal{W}$ and $\mathcal{G}$ be the (indices) of **wires** and **gates** of C, respectively.
- For $w \in \mathcal{W}$, associate a pair of random 'keys" $k_w = (k_w^0, k_w^1) \in (\{0,1\}^n)^2$.
- For $g \in \mathcal{G}$ with input wires i and j, and output wire h, let $T(g)$ be the following table:

input wire i	input wire j	output wire h	hidden output wire
k_i^0	k_j^0	$k_h^{g(0,0)}$	$E_{k_i^0}(E_{k_j^0}(k_h^{g(0,0)}))$
k_i^0	k_j^1	$k_h^{g(0,1)}$	$E_{k_i^0}(E_{k_j^1}(k_h^{g(0,1)}))$
k_i^1	k_j^0	$k_h^{g(1,0)}$	$E_{k_i^1}(E_{k_j^0}(k_h^{g(1,0)}))$
k_i^1	k_j^1	$k_h^{g(1,1)}$	$E_{k_i^1}(E_{k_j^1}(k_h^{g(1,1)}))$

Figure: Table for gate g, with input wires i and j, and output wire h.

The Garbled Circuit, cont.

input wire i	input wire j	output wire h	hidden output wire
k_i^0	k_j^0	$k_h^{g(0,0)}$	$E_{k_i^0}(E_{k_j^0}(k_h^{g(0,0)}))$
k_i^0	k_j^1	$k_h^{g(0,1)}$	$E_{k_i^0}(E_{k_j^1}(k_h^{g(0,1)}))$
k_i^1	k_j^0	$k_h^{g(1,0)}$	$E_{k_i^1}(E_{k_j^0}(k_h^{g(1,0)}))$
k_i^1	k_j^1	$k_h^{g(1,1)}$	$E_{k_i^1}(E_{k_j^1}(k_h^{g(1,1)}))$

Let $\mathcal{I}$ and $\mathcal{O}$ be the input and outputs wires of C.

- For $g \in \mathcal{G}$, let $\widetilde{T}(g)$ be a random permutation of the fourth column of $T(g)$.
- For $w \in \mathcal{W}$, let $\mathrm{C}(x)_w$ be the **bit-value** computation of $\mathrm{C}(x)$ assigns to w
- Given
  1. $\widetilde{T} = \{(g, \widetilde{T}(g))\}_{g\in\mathcal{G}}$.
  2. $\{(w, k_w^{\mathrm{C}(x)_w})\}_{w\in\mathcal{I}}$ for some x.
  3. $\{(w, k_w = (k_w^0, k_w^1)\}_{w\in\mathcal{O}}$.

  One can efficiently compute $\mathrm{C}(x)$.
- (essentially) The above leaks no additional information about x!

Example, GV for OR

On board...

The protocol

- Let $f(x_A, x_B) = (f_A(x_A, x_B), f_B(x_A, x_B))$ be a function, and let C be a circuit that computes f.
- Let $\mathcal{I}_A$ and $\mathcal{I}_B$ be the input wires corresponds to x_A and x_B respectively in C, and let $\mathcal{O}_A$ and $\mathcal{O}_B$ be the output wires corresponds to f_A and f_B outputs respectively in C.
- Recall that $C(x)_w$ is the bit-value the computation of $C(x)$ assigns to w.
- Let (S, R) be a secure protocol for OT.

Protocol 12 ((A, B))

Common input: 1^n. A/B**'s input:** x_A / x_B

1. A: Sample at random $\{k_w = (k_w^0, k_w^1)\}_{w \in \mathcal{W}}$, and generate $\widetilde{T}$.
2. A: Send $\widetilde{T}$, $\{(w, k_w^{C(x_A,\cdot)_w})\}_{w \in \mathcal{I}_A}$, and $\{(w, k_w)\}_{w \in \mathcal{O}_B}$ to B.
3. $\forall w \in \mathcal{I}_B$: The parties interact in $(S(k_w), R(C(\cdot, x_B)_w))(1^n)$.
4. B: Compute the (garbled) circuit, and send $\{(w, k_w^{C(x_A,x_B)_w})\}_{w \in \mathcal{O}_A}$ to A.
5. The parties compute $f_A(x_A, x_B)$ and $f_B(x_A, x_B)$ respectively.

Example, protocol for OR

On board...

Claim 13

Protocol 12 securely computes f (in the semi-honest model)

Proof: We focus on the security of A. For a semi-honest B′, define

Algorithm 14 ($\widehat{\mathsf{B}}$)

input: 1^n and x_{B}.

1. Send x_{B} to the trusted party, and let o_{B} be its answer.
2. Emulate the first 4 steps of $(\mathsf{A}(0^{|x_{\mathsf{A}}|}), \mathsf{B}'(x_{\mathsf{B}})(1^n))$.
3. For each $w \in \mathcal{O}_{\mathsf{B}}$: permute the order of the pair k_w according to o_{B}, and the key of w computed in the emulation.
4. Complete the emulation, and output the output that B′ does.

Claim: $\widehat{\mathsf{B}}$ is a good “simulator" for B′.

Security of B ?

Extensions

- Efficiently computable f

  Both parties first compute C_f – a circuit that compute f for inputs of the right length

- Hiding C? All but its size

Section 5

Malicious Security

Semi-honest to malicious security

Let $\pi^h = (\mathsf{A}^h, \mathsf{B}^h)$ semi-honest protocol for f. The parties prove that they act "according to π^h":

1. Forces the parties to chose their random coin properly
2. Before each step, the parties prove in $\mathcal{ZK}$ that they followed the prescribed protocol (with respect to the random-coins chosen above)

We will use

1. Perfectly binding POK. NI commitment Com. (?)
2. Zero-knowledge proof for (soon to be implicitly defined) $\mathcal{L} \in \mathcal{NP}$ with negligible soundness and completeness error
3. Let $\ell(n)$ bound the number of random coins the parties of π^h use on common input of length n.

The compiler

Protocol 15 (Maliciously secure protocol $\pi = (\mathsf{A}, \mathsf{B})$)

Common input: x_c, let $n = |x_c|$

P**'s input:** x_{P}.

1. For each $\widehat{\mathsf{P}} \in \{\mathsf{A}, \mathsf{B}\}$:
   - **1.1** $\widetilde{\mathsf{P}}$ commits to x_{P}.
   - **1.2** $\widetilde{\mathsf{P}}$ commits to $r^1_{\mathsf{P}} \leftarrow \{0,1\}^\ell$.
   - **1.3** Other party sends $r^2_{\mathsf{P}} \leftarrow \{0,1\}^\ell$ to $\widetilde{\mathsf{P}}$
   - **1.4** $\widetilde{\mathsf{P}}$ sets $r_{\mathsf{P}} = r^1_{\mathsf{P}} \oplus r^2_{\mathsf{P}}$.
2. The parties interact $(\mathsf{A}^h(x_{\mathsf{A}}; r_{\mathsf{A}}), \mathsf{B}^h(x_{\mathsf{B}}; r_{\mathsf{B}}))(x_c)$:
   - **2.1** After party $\widetilde{\mathsf{P}}$ sends a message m, it proves in ZK that m is what P^h would send on input x_{P}, randomness r_{P}, and previous messages received.
   - **2.2** The other party abort, if proof fails.

Claim 16

Assume π^h is semi-honest secure for f, then π is maliciously secure for f (w/ abort).